intel®

APPLICATION NOTE

AP-7

Disk Controller Design Uses New Bipolar Microcomputer LSI Components

Glenn Louie
Applications Research Engineer

98-031A

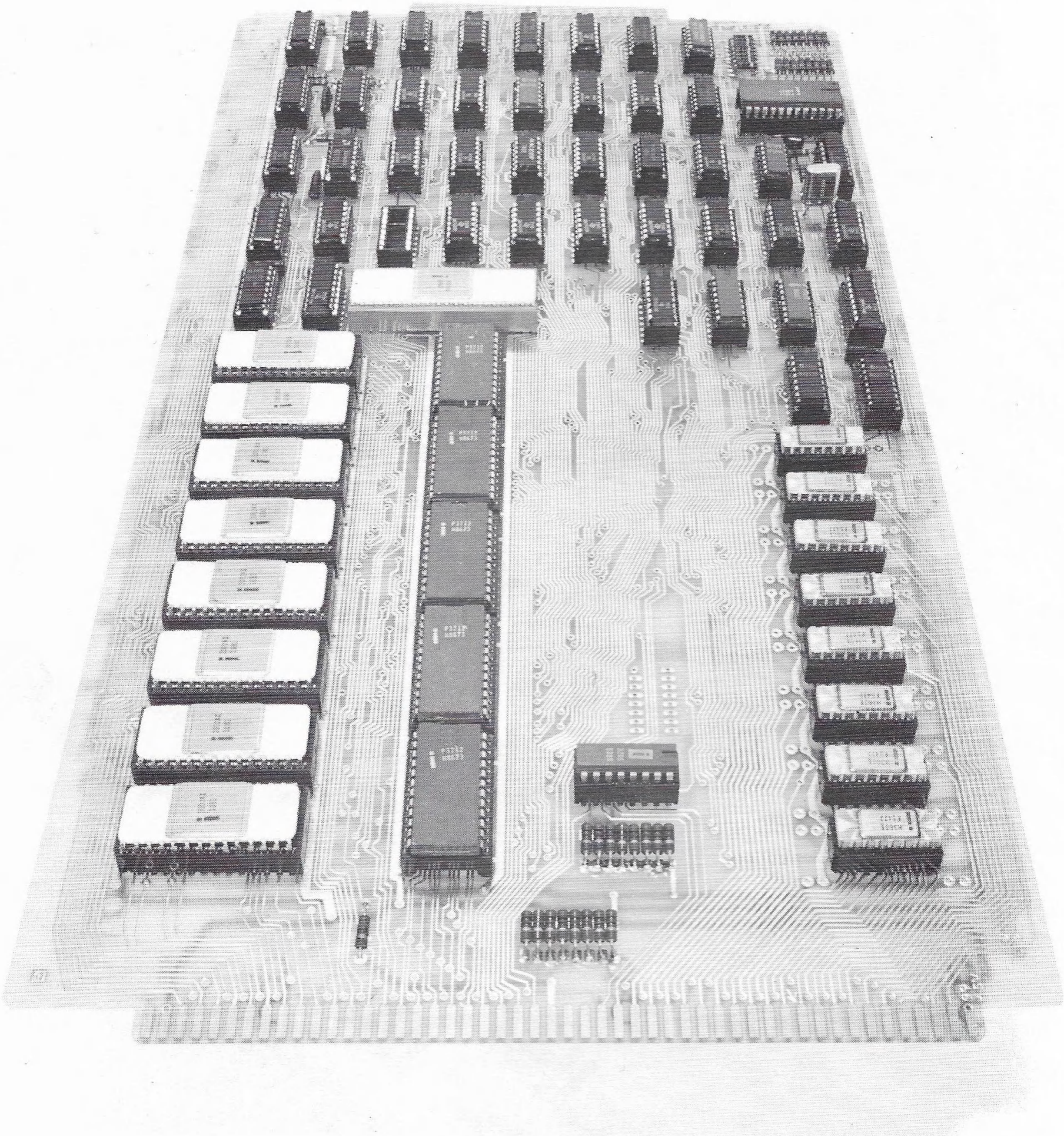


Figure 1. Bipolar Microprogrammed Disk Controller

Disk Controller Designed With Series 3000 Computing Elements

With the introduction of the first microprocessor, digital designers began a massive switch to programmable LSI technology, away from hardwired random logic. Designers found that with these new LSI components and the availability of low cost ROMs they could easily implement structured designs which were both cost effective and flexible. However, not all digital designs were amenable to the microcomputer approach. One of the basic limitations was the speed at which a particular critical program sequence could be executed by a microprocessor. The early P-channel MOS microprocessors, such as Intel's 4004 and 8008, were able to solve a broad class of logic problems where speed was not essential. With the introduction of the more powerful n-channel MOS microprocessors, such as the Intel 8080, the range of applications was significantly broadened, but there still existed a class of applications that even these newer devices were not fast enough to handle.

Recently, two new Schottky bipolar LSI computing elements, members of the Intel Bipolar Microcomputer Set, were introduced which expand the range of microcomputer applications to include high speed peripheral controllers and communication equipment. The new elements are the 3001 Microprogram Control Unit (MCU) and the 3002 Central Processing Element (CPE). These two components facilitate the design of specialized, high speed microprocessors that together with a minimum of external logic perform the intricate program sequences required by high speed peripheral controllers.

A multi-chip bipolar microprocessor differs from the single chip MOS microprocessor in that the bipolar microprocessor is programmed at the microinstruction level rather than at the macroinstruction level. This means that instead of specifying the action via a macroprogram using a fixed instruction set, a designer can specify the detailed action occurring inside the microprocessor hardware via a microprogram using his own customized microinstructions.

In general, microinstructions are wider than macroinstructions (e.g. 24 to 32 bits) and have a number of independent fields that specify simultaneous operations. In a single microcycle, an arithmetic operation can be executed while a constant is stored into external logic and a conditional jump is being performed.

A bipolar LSI microprocessor design is similar to a general MSI/SSI microprocessor design where the intricacies of the application are imbedded in the program patterns in ROM. However, the large amount of logic necessary to access the microcode has been replaced by the LSI MCU chip. Also,

the MSI logic required to provide the arithmetic and register capabilities has been replaced by the functionally denser LSI CPE slices. Because of these new LSI chips, microprogramming with all its advantages can now be applied to designs which previously were unable to justify microprogramming overhead.

The effectiveness of these new LSI components in a high speed peripheral controller design has been demonstrated by the Applications Research group at Intel with the design of a 2310/5440 moving head disk controller (BMDC). The BMDC has a total of 67 IC chips and is packaged on a printed circuit board measuring 8" x 15", as shown in Figure 1. Disk controllers of equivalent complexity realized with conventional components typically require between 150 and 250 I.C.'s. The BMDC performs all the operations required to interface up to four "daisy chained" moving head disk drives, with a combined storage capacity of 400 megabits, to a typical minicomputer. It is fast enough to keep up with the drive's 2.5 MHz bit serial data stream while performing the requisite data channel functions of incrementing an address register, decrementing a word count register, and terminating upon completion of a block transfer.

The BMDC interacts with the minicomputer's disk operating system (DOS) via I/O commands, interrupts and direct memory access (DMA) cycles. The I/O commands recognized by the BMDC's microprogram are:

Conditions In
Seek Cylinder
Write Data
Read Data
Verify Data
Format Data

The BMDC sends an interrupt to the minicomputer when either a command is successfully executed, a command is aborted, or a drive has finished seeking. The DOS then interrogates the BMDC with a Conditions In command. The following flags specify the conditions which the BMDC can detect:

Done flag
Malfunction flag
Not Ready flag
Change In Seek Status flag
Program Error flag
Address Error flag
Data Error flag
Data Overrun flag

Data transfers between the minicomputer and the disk BMDC occur during DMA cycles. DMA cycles are also used for passing command information from the minicomputer to the BMDC.

The bipolar LSI microcomputer in the BMDC performs the necessary command decoding, address checking, sector counting, overlap seeking, direct memory accessing, write protection, password protection, overrun detection, drive and read selection, and formatting. External hardware assists the microprocessor in updating the sector counter, performing parallel-to-serial and serial-to-parallel conversion, and generating the CRC data checking information. The BMDC uses a special purpose microprocessor, configured with the components listed in Table A. The LSI microprocessor uses an MCU, an 8:1 multiplexer, eight 3601 PROMs, a command latch, a data buffer, and an array of eight CPE slices (Fig. 2). The characteristics of this design, only one of many possible with the 3000 family, are as follows:

- 400 nsec system clock
- 16-bit wide CP array
- Ripple carry CPE configuration
- Non-pipelined architecture
- One level subroutining
- 230 32-bit microinstructions
- Word to 4-bit nibble serialization

The MCU controls the sequence in which microinstructions are executed. It has a set of unconditional and conditional jump instructions which is based on a 2-dimensional array for the microprogram address spece called the MCU Jump Map.[1]

PART #	DESCRIPTION	QUANTITY
3001	MCU	1
3002	CPE	8
3212	8 bit I/O Port	6
3205	1 of 8 Decoder	2
3601	1K PROM	8
3404	6 bit Latch	1
74173	4 bit Gated D F/F	1
74174	6 bit D F/F	1
74175	4 bit D F/F	1
74151	8:1 Multiplexer	1
8233	Dual 4:1 Multiplexer	2
9300	4 bit Shift Register	1
9316	4 bit Binary Counter	1
8503	CRC Generator	1
7474	Dual D F/F	5
7473	Dual J-K F/F	2
7451	And-Or-Invert Gate	1
7404	Hex Inverter	6
7400	Quad 2 Input Nand Gate	9
74H08	Quad 2 Input And Gate	1
7403	Quad 2 Input Nand O.C. Gate	2
7438	Quad O.C. Drivers	4
74H103	Dual J-K F/F	2
	Total	67 I.C. Packages

Table A. I.C. Component List for Disk Controller

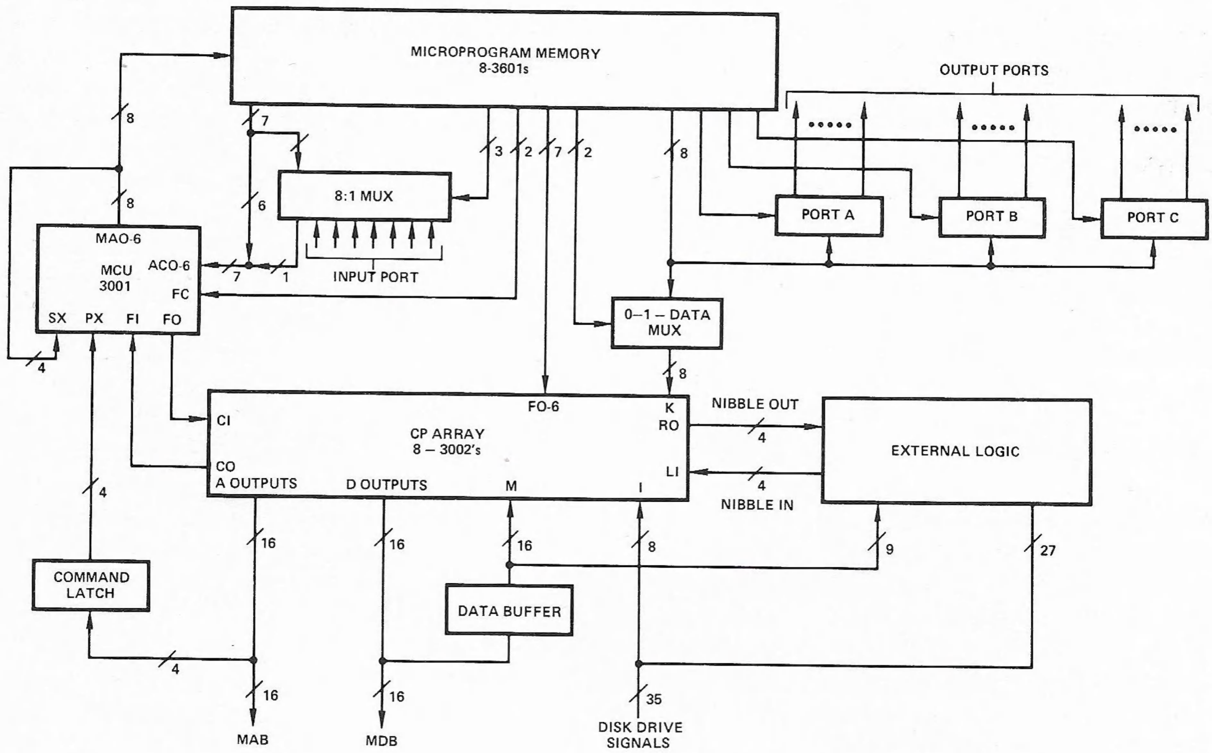


Figure 2. Disk Controller – The various elements of a specialized microprogrammed processor is shown with the external logic which together is the entire disk controller.

In addition, the MCU is connected in such a manner as to perform command decoding, external input testing, and one level subroutining.

Command decoding is achieved by connecting the command latch to the Primary Instruction (PX) bus inputs and using the JPX instruction (Fig. 3). The testing of external input signals is performed by routing the least significant bit (LSB) of the seven bit jump code through an eight-to-one multiplexer (Fig. 2). The multiplexer is controlled by a 3-bit Input Select Code which selects either the LSB of the jump code or one of 7 external input signals to be routed to the MCU. This technique has the effect of conditionally modifying an unconditional jump code so that the next address will either be an odd or even location (Fig. 3). A one instruction wait for external signal loop can be simply implemented in this fashion.

One level subroutining is achieved by feeding the four least significant bits of the address microprogram outputs back into the secondary instruction (SX) inputs. Enough program status information can then be saved in the internal PR latch when a subroutine is called with a JPX instruction so that upon exiting, a subroutine with a JPR instruction, control can be returned to the procedure which called it (Fig. 4). This technique saves a significant amount of microcode in the BMDC because some long sequences do not have to be repeated.

The microprogram control store is an array of eight 3601 PROMs organized to give 256 words x 32 bits (230 words were required for the BMDC). The 32-bit wide word is divided into the following subcontrol fields:

1. Jump Code field	7 bits
2. Flag Control field	2 bits
3. CPE Function field	7 bits
4. Input Select field	3 bits
5. Output Select field	3 bits
6. Mask or Data field	8 bits
7. Mask Control field	2 bits
TOTAL	32 bits

The command latch and data buffer retain command information from the computer so that the memory bus will not be held up if the BMDC should be busy performing an updating task. The data buffer also retains the next data word during a Write Data to disk operation.

The CP array is connected in a ripple carry configuration as shown in Figure 5. The eight CPE slices provide the BMDC with a 16-bit arithmetic, logic and register section. Word to nibble serialization is made possible by connecting the Shift Right Outputs (RO) of the first, third, fifth, and seventh CPE to the Nibble Out bus. By using only four shift right operations a word in a register can be converted into four 4-bit nibbles. The final serialization of these nibbles is done in the external logic. Similarly, the Shift Right Inputs (LI) of the second, fourth, sixth, and eighth CPE are connected to the Nibble In bus so that with only four shift right operations, a word can be assembled from four nibbles.

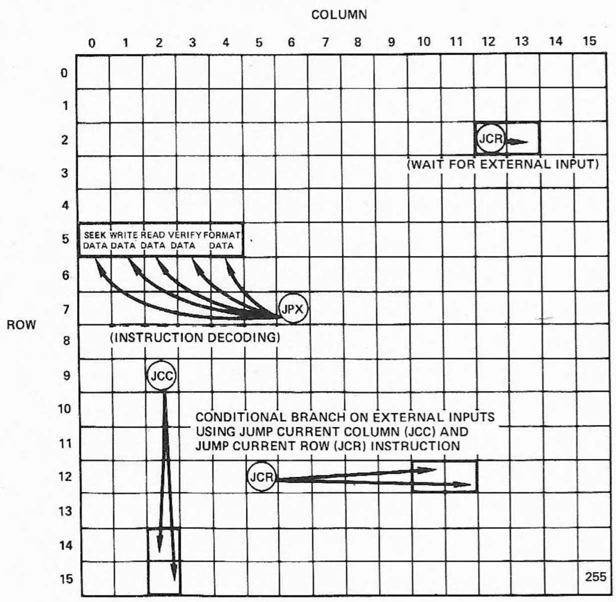


Figure 3. MCU Jump Map for instruction decoding and conditional branching on external inputs

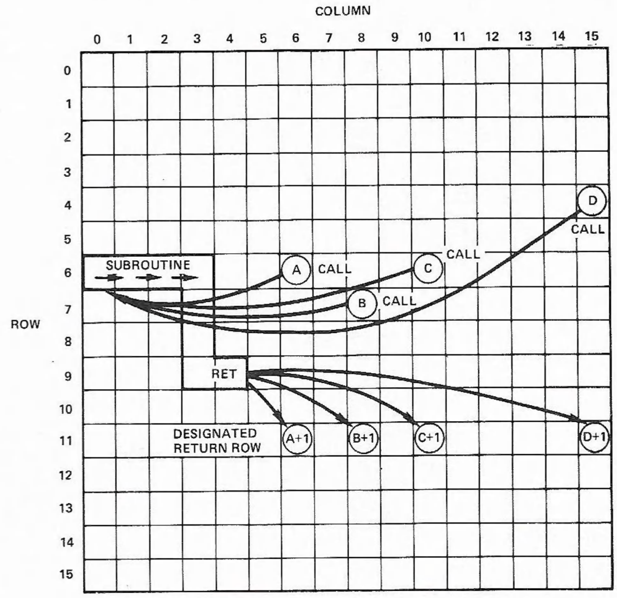


Figure 4. MCU Jump Map for one level subroutine call and return. A subroutine is called from four different places in the program each with a unique column number. Upon returning from the subroutine, control will be transferred back to the portion of program which called it. A subroutine may be called from a maximum of 16 different places.

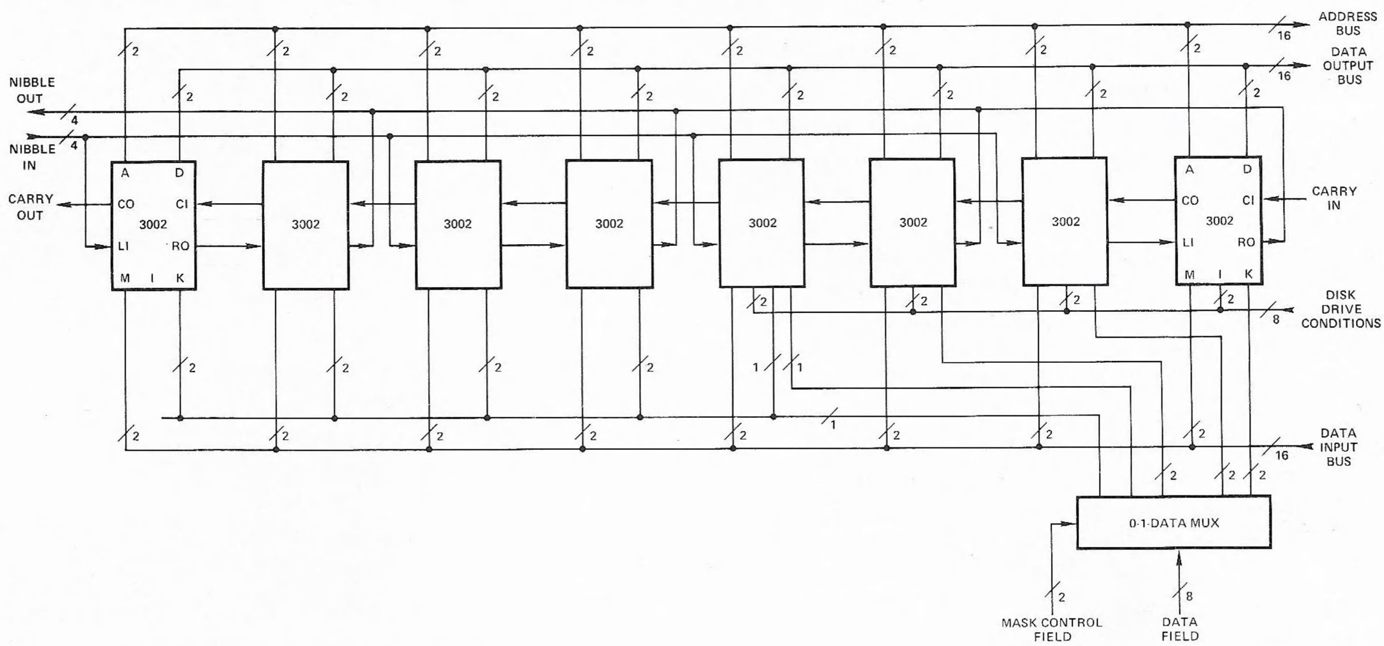


Figure 5. CPE Array – A 16-bit arithmetic, logic and register section is built up with 8 CPE slices connected in a ripple carry configuration. The K, I, and M bus is used for loading information into the CPE slices. The LI inputs and RO outputs are connected to make up the Nibble In and Nibble Out buses.

An eight bit mask bus is connected to the mask inputs of the least significant half of the array. The mask inputs of the most significant half of the CP array are all tied to the eighth mask bit. A constant with a value between +127 and −128 can therefore be loaded into the array from the microprogram. The mask bus comes from the data field of the microprogram via a 0-1 data multiplexer. When the CP array requires either an all one or all zero mask, the data field is freed to provide data to external logic.

The 3002 CPE is an extremely flexible component which makes it particularly attractive for controller designs. The Memory Address Register makes an ideal DMA address register.(1) The accumulator (AC) register, which also has its own output bus can be used as a data word buffer during a write DMA cycle. Concurrently, another word can be assembled in the T register using the shift right operation. The three separate input buses provide a multiplexing capability for routing different data into the CPE. In the BMDC, the I-bus is used for loading disk drive conditions, the K-bus for loading mask or constant information, and the M-bus for reading an external data buffer. The arithmetic logic section performs zero detection and bit testing with the result delivered to the MCU chip via the carry out line. Finally, the eleven scratchpad registers allow the controller to retain data and status for the processor.

The CP array in the microprocessor performs the following for the BMDC with its registers and arithmetic functions.

1. Sector counting
2. Word to nibble serialization
3. Drive seek status monitoring
4. Header checking
5. DMA address incrementing
6. Word counting
7. Multi-sector length counting
8. Automatic resynchronization of sector counter
9. Accessing of additional information from memory
10. Time delays

The organization of the microprocessor was chosen to maximize the use of the MCU and CPE in performing the various tasks required for disk control. However, there are some specialized tasks which are more economically performed by external logic. The microprocessor controls this external logic by output ports which are selected by the output select field in the microinstruction. The

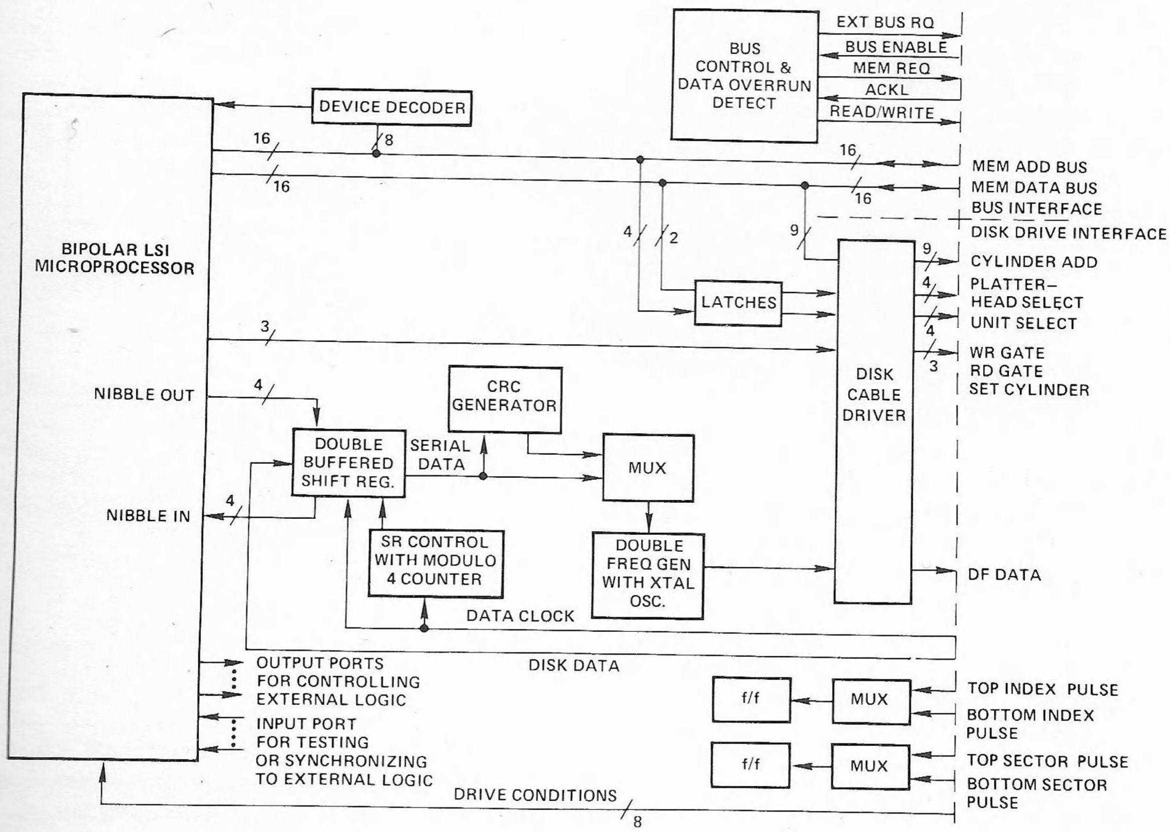


Figure 6. **External Logic – Microprocessor monitors and controls external logic via input-output port to perform specialized disk controller functions.**

data to these ports is delivered from the shared data field.

The external logic section of the BMDC (Fig. 6) has a double buffered 4-bit shift register which is used for initial packing and the final serialization of data. It is controlled by a modulo-4 counter circuit. During a write operation, serial data from the shift register is encoded by the clock controlled double frequency encoder and sent to the drive. As data is being transferred to a cyclic redundancy code (CRC) is generated and then appended to the end of the data stream to be recorded on the disk. The external logic also contains addressing latches and flag flip-flops to capture sector and index pulses. It also contains main memory bus control circuitry for performing bus protocol, bus acquisition, and data overrun detection.

The microprogram for the BMDC microprocessor directly implements the six I/O commands. The program controls the sequential action of the various elements of the microprocessor and of the external logic needed to decode and execute the commands. In Figure 7, the flow chart of the Read command shows the actions required to read a file off the disc. The BMDC first selects the drive specified by the command and checks its ready status. It then uses a memory pointer passed to it by the command to access four more words from the main memory using DMA cycles. The first word is the Header, which contains the track address and sector address information. The second word is the Starting Address specifying the first location in memory where the data is to be stored. The third word is the Block Length of the file to be retrieved. All of the address information and the Block Length are stored in several CPE registers for further processing. The fourth word is the Password which is compared against a microprogram word to insure that the command from the computer is a valid one and not a program error. The password can prevent an erroneous command, due to a user programming error, from destroying important files on the disc.

After the password check, the BMDC resynchronizes the sector counter if necessary and waits for the desired sector by monitoring the sector pulse flag. When the desired sector arrives, the BMDC synchronizes itself to a start nibble and reads the header which it compares to the desired header to insure that the head is positioned properly. It then reads and stores 128 words of data at sequential locations in memory. A cyclic redundancy code is compiled during the read oper-

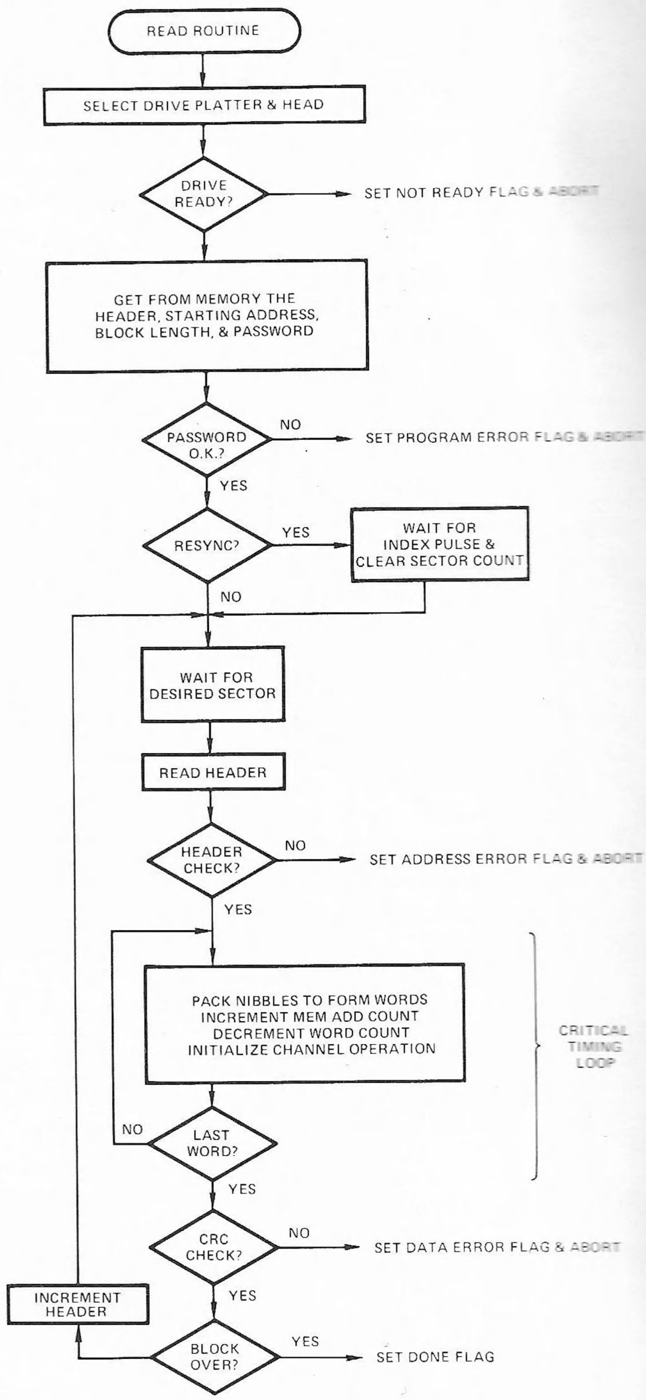


Figure 7. Read Command Flowchart — This flowchart is coded in the microprogram which when executed performs the disk Read operation.

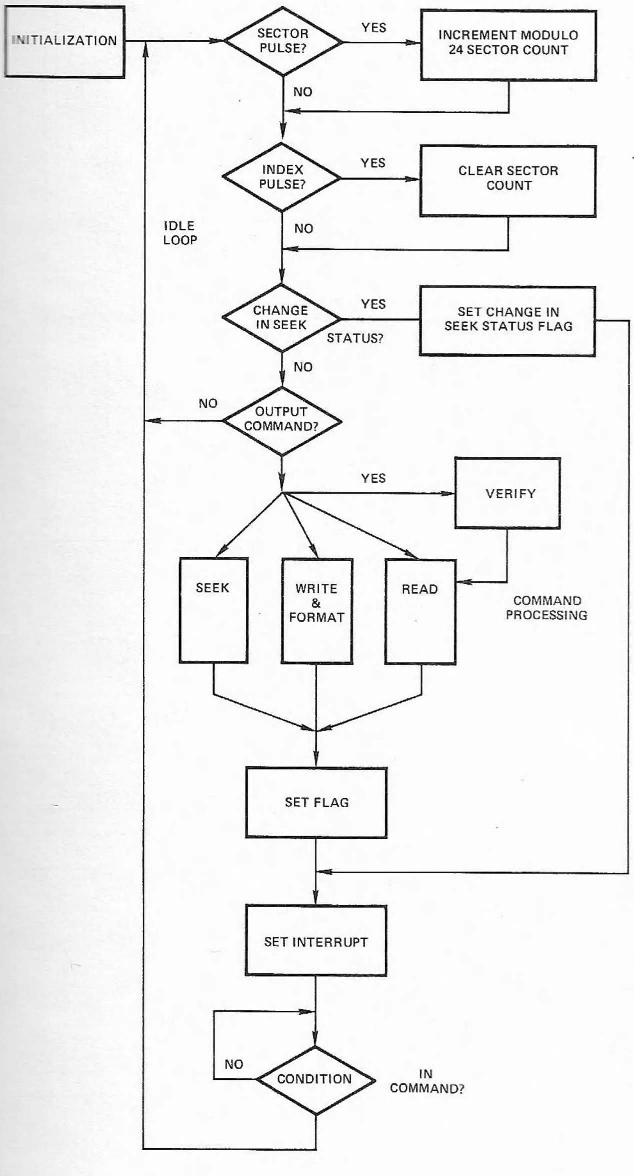


Figure 8. **BMDC Flowchart – The BMDC runs in the idle loop when it is not busy doing command processing.**

ation and compared against the CRC word read in after the data. At the end of each sector the block length is decremented to see if it is the last sector. If it is not, the sector address is incremented and another sector is read.

In addition to the command routines, the microprogram has an idle loop routine (Fig. 8) which the BMDC executes when it is not busy with a command. While in the loop, the BMDC updates the sector count, monitors the drives seeks status lines and decodes any disc commands from the disc operating system in the minicomputer.

The design process for the BMDC began with an evaluation of what disc controller operations could effectively be handled by the microprocessor. This also determined what had to be performed by external logic. A microprocessor configuration was then established and certain critical sequences were programmed to verify that the configuration was fast enough. A flow chart was produced and the microprogram coded directly from it. All attempts were made to use the MCU and CPE slices effectively and keep the microprogram within 256 words. The assignment of MCU addresses which initially appeared difficult, was, with a little experience, quite straight forward and less restrictive than a state counter design. After the coding, the microprogram was assembled and loaded into the microprocessor's control memory.

The BMDC design demonstrates how a specialized high speed microprocessor can be designed using standard bipolar LSI devices and microprogrammed to perform disc control functions with the addition of a small amount of external logic. The flexibility of Series 3000 allows a designer to optimize the configuration for his application. For extremely high speed applications, the designer can add fast carry logic and microinstruction pipelining to his microprocessor to achieve a 150 nsec 16-bit microprocessor.

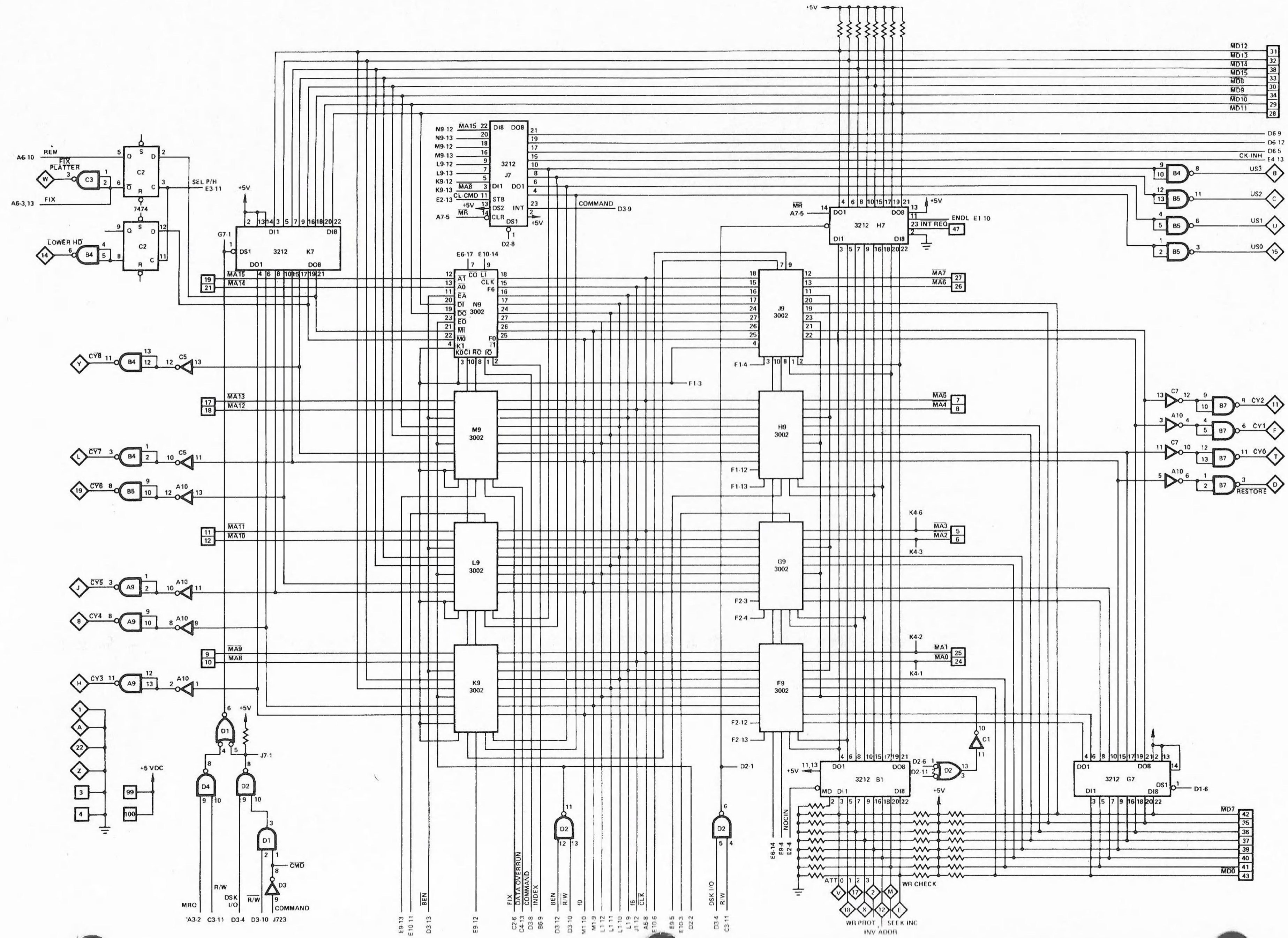
At Intel, our design experience with the BMDC design exercise has shown that the use of the MCU and CPE results in a clean, well structured design. The complexity of the design resides primarily in the microprogram leaving the external logic relatively simple. During debugging, most of the problems encountered were restricted to the microprogram which was easily modified and debugged using bipolar RAM for the control memory.

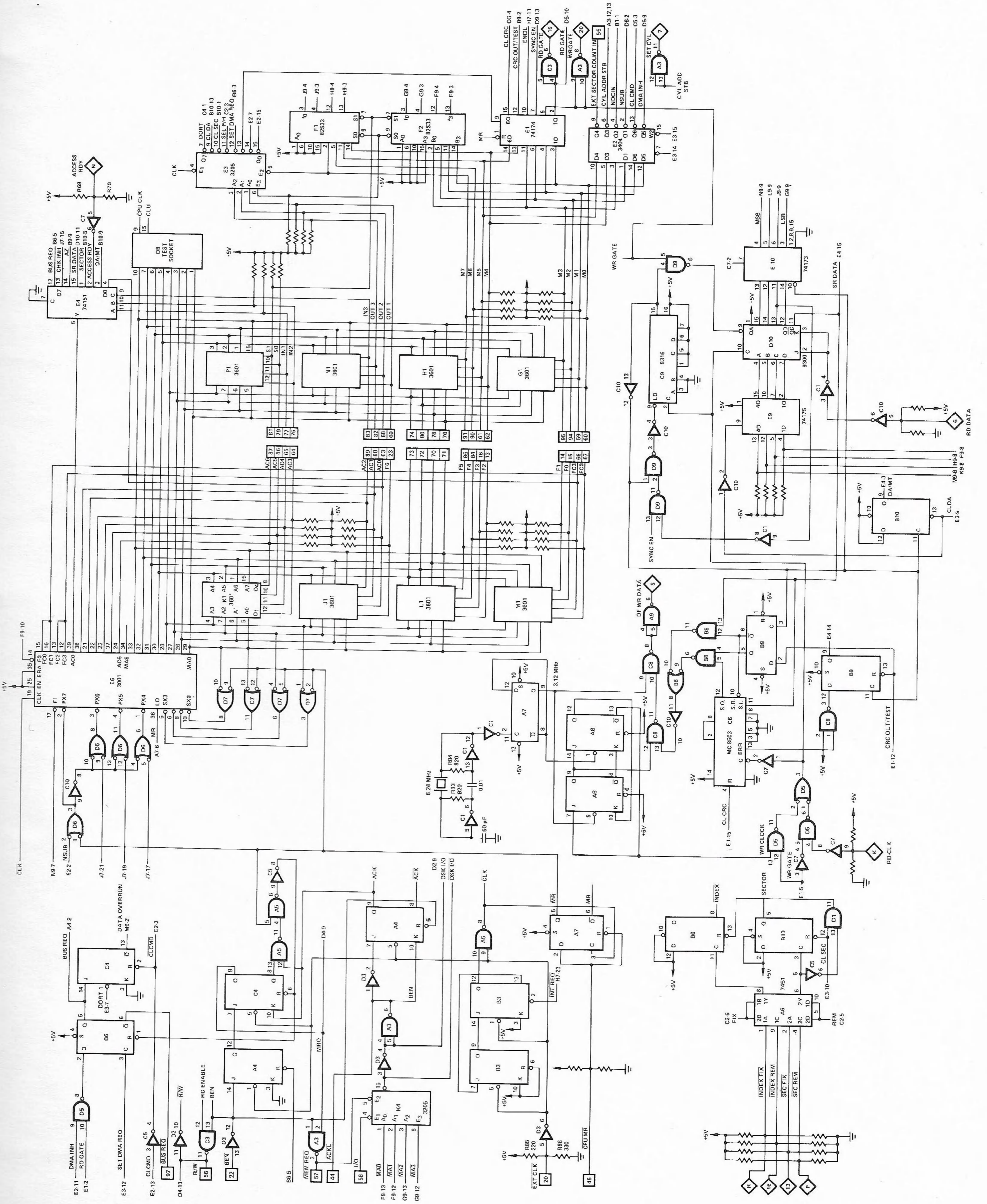
References

1. J. Rattner, J. Cornet, M. E. Hoff, Jr., "Bipolar LSI Computing Elements Usher In New Era of Digital Design," ELECTRONICS, September 5, 1974, pp 89–96.

APPENDIX—DISK CONTROLLER SCHEMATICS

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.





intel®

January 1975, INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) 246-7501

Printed in U.S.A. MCS-398-0175/27.5K